* 1. Text Preprocessing (10%)

1. How do you choose the tokenizer for this task? Could we use the white space to tokenize the text? What about we use the complicated tokenizer instead? Make some discussion. (5%) (You might want to explain it by showing the performance comparison with different tokenizer. If you are not familiar with tokenizer, check Here.

2. Why we need to use special token like <pad> and <unk>? (2%)

3. Briefly explain how your procedure handles the text data. (3%) (tokenize, stop word, min count setting, etc)

1. 這裡使用的tokenizer是keras.preprocessing.test裡的函數，是直接把句子中的每個詞，依照空白區隔間隔一個字，把每個字轉成一個向量表示。可以使用空白切分字詞。除了將句子切分為單詞，也可以在相關性較高的連續字詞中切分，這時的token不一定為一個單詞，應該會有更好的效果。
2. 因為句子的長度不一定都一樣，所以轉換成向量的時候長度也會不一致，需要透過padding把向量補0或調整成一樣長度的向量，句子的最大長度要視放入的資料決定。
3. 我的資料預處理是先把title的部分用上述的tokenizer把每個字都轉成對應向量，再把訓練的最大字數設為20(因為看大多句子數都在10個字左右)，其他不足的地方補零。在把label的文字用LabelEncoder處理成對應的index。
   1. RNN (30%) Build the recurrent neural network to solve this task and answer the following question

1. Outperform the RNN baseline on the Kaggle in-class competition. (20%)

2. How do you choose the initial embedding for this task? Why we often use the pretrained embedding instead of random initialization? (5%)

3. Discuss the model structure (e.g. RNN, GRU or LSTM) you design. (5%)

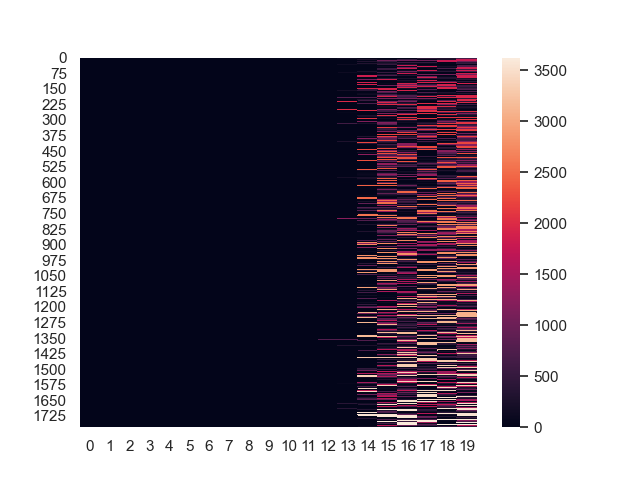
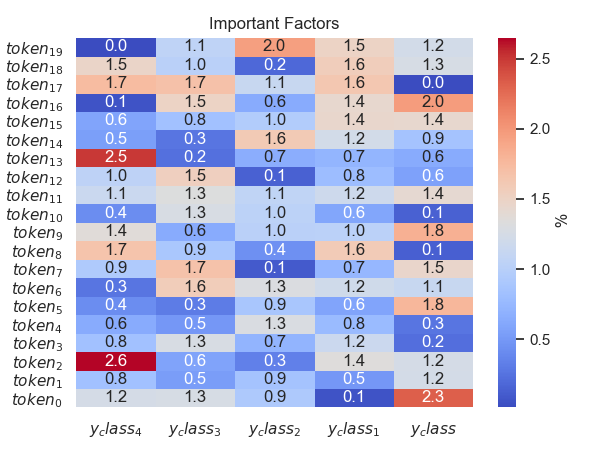
我直接用0!!!!!!!!!!!!!!!!!!!!!!一般用pretrain model可以學習到比較多的單詞，別人訓練的詞袋也會比較大，訓練的model會更robust!!!!但很可惜我還沒用。我這裡主要是用LSTM，(embedding=>dropout 0.3 =>LSTM,100neurons => dense layer, 1024 neurons => dropout 0.8 => dense layer, 1024 neurons => dropout 0.8 => dense layer, 5 neurons for 5 classes.)

* 1. Transformer (30%) Build the transformer to solve this task and answer the following question

1. Outperform the transformer baseline on the Kaggle in-class competition. (20%)

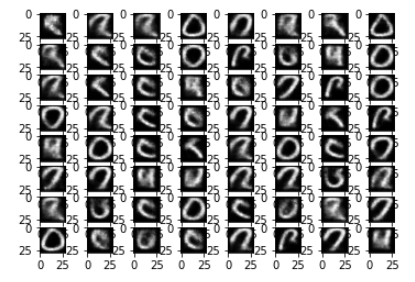
2. Please show the attention map in the last layer of transformer of some examples to find out which token contributes higher attention in the classification result. Does your finding make sense? Please make some discussion. (5%)

3. Discuss the model setting (e.g. the hyperparameters in transformer or the input formation) you design. (5%)

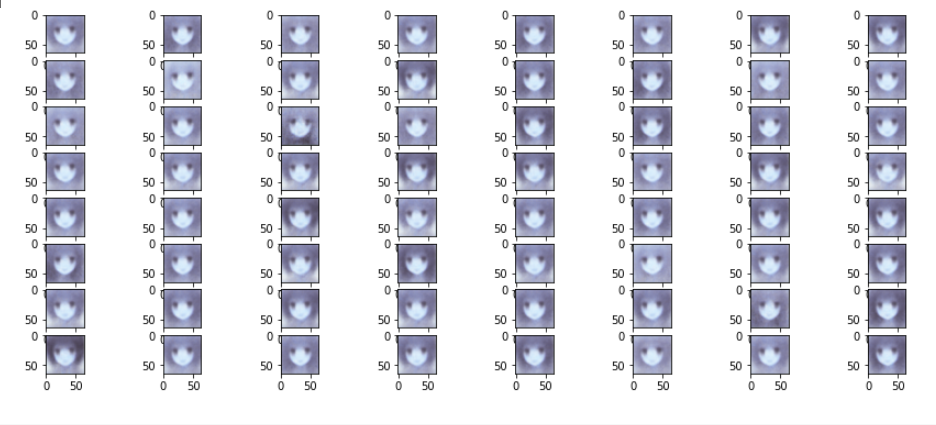


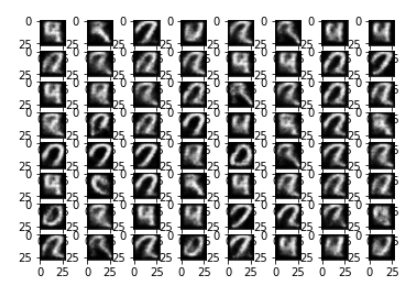
2.1

1. Implement VAE and show the learning curve and some reconstructed samples like the given examples. (10%)



2. Sample the prior p(z) and use the latent codes z to synthesize some examples when your model is well-trained. (5%)





3.Show the synthesized images based on the interpolation of two latent codes z between two real samples. (5%)





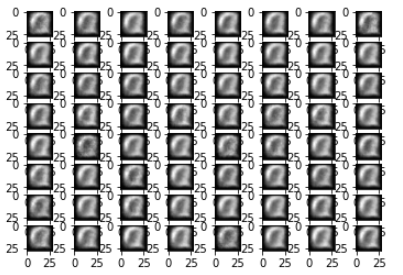
1. Multiply the Kullback-Leiblier (KL) term with a scale λ and tune λ (e.g. λ = 0 and λ = 100) then show the results based on steps 1, 2, 3 and some analyses. (10%)

Total loss = 50% rec + 50% KL divergence loss

當reconstruction loss(λ = 0)比重較大時會較靠近原始圖片

反之則較靠近sample出來的分布所以較不清楚

λ = 100



λ = 0

